

AUSTRALIAN OS9 NEWSLETTER

sides 'No. of cylinders' (in decimal) :Interleave value: (in decimal) @FREE Syntax: Free [devname] Usage : Displays number of free sectors on a device @GFX Syntax: RUN GFX(<funct><args>) Usage : Graphics interface package for BASIC09 to do compatible VDG graphics commands @GFX2 Syntax: RUN GFX2([path]<funct><args>) Usage : Graphics interface package for BASIC09 to handle

Usage : EDITOR Gordon Bentzen (07) 344-3881

help to SUB-EDITOR Bob Devries (07) 372-7816

from OS TREASURER Don Berrie (07) 375-1284

single line LIBRARIAN Jean-Pierre Jacquet (07) 372-4675

directory @INKEY Fax Messages (07) 372-8325

input a SUPPORT Brisbane OS9 Users Group

the pro memory text files @LOAD Syntax: Load <pathname> [L] Usage : Loads modules into memory @MAKDIR Syntax: Makdir <pathname> Usage : Creates a new directory file @MDIR Syntax: Mdir [e] Usage : Displays the present memory module directory Opts : e = print extended module directory @MERGE Syntax: Merge <path>

@MFREE Synt

@MODPATCH

memory from c

compare modul

to module C o

module M = ma

Usage : Set mo

monochrome m

and links an OS

Procs [e] Usage

display all pro

current data directory path

execution directory path

Usage : Gives the file or directory a new name

module> Usage : BASIC09 run time package

[yy/mm]

Syntax:

num @

@TMODE Syntax: Tmode [pathname] [params] Usage: Displays or changes

the operating parameters of the terminal

@TUNEPOR

[value] Adjust the baud value for the serial port

@UNLINK Syntax: Unlink

<modname> Usage : Unlinks module(s) from memory

@WCREATE Syntax:

Addresses for Correspondence

Editorial Material:

Gordon Bentzen
8 Odin Street
SUNNYBANK Qld 4109

Subscriptions & Library Requests:

Jean-Pierre Jacquet
27 Hampton Street
DURACK Qld 4077

Volume 5

September 1991

Number 8

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group
Volume 5 Number 8

EDITOR : Gordon Bentzen
SUBEDITOR : Bob Devries

TREASURER : Don Berrie
LIBRARIAN : Jean-Pierre Jacquet

SUPPORT : Brisbane OS9 Level 2 Users Group.

This edition is the first for our FOURTH year of support to the OS9 community and we thank the 33 members who have already re-subscribed for the coming year. This edition will also be mailed to all members on our mailing list as at August 1991 just in case you have forgotten to mail your cheque or the mail has been slow. We have received a letter from a couple of members advising that they will not be re-subscribing this year and we do appreciate this courtesy, thankyou.

UGCAT. Several copies of our P.D. library catalogue "UGCAT" (UserGroup Catalogue) have been requested and subsequently mailed. We understand, however, that some of these were not readable in 40 track drives. Under OS9 we have the ability to write to 40 track double sided disks in 80 track drives provided that the drive can be switched to 48 T.P.I. mode. Many 40 track disks have been produced using this method, and in most cases it works O.K. but not always, apparently.

To overcome this problem, our librarian has added a 40 track drive to his system. If you have trouble reading any P.D. material, please return your disks to the librarian, Jean-Pierre Jacquet, and they will be redone at no extra charge.

The UGCAT database has now been extended to include archives on Disk #11, and we will update your copy of UGCAT at no charge if requested. Please include postage stamps for return postage.

Please send all requests for P.D. archives and UGCAT to our librarian, particularly if using 35 or 40 track disk format. A copy charge of \$2 per disk applies to any disk format.

FLOPPY DISKS. We have noted that quite a few members have included TANDY disks with their request for P.D. files, and we would like to point out that disks can be purchased at a much lower price than Tandy charge. Prices should be in the 50 cents to 60 cents per disk. We have purchased disks in lots of 50 for \$20 at times and have found these to be quite reliable when formatted at 720k (80 tracks double sided). If you are not able to purchase disks at reasonable

prices, please contact us and we will be able to obtain some for you.

Although we do not wish to become suppliers of hardware or software, we will offer this as a non-profit service to members in need.

MM-1. Last month's edition contained a rather lengthy article on the MM-1 by Paul Ward of Interactive Media Systems, so we won't go into that again here, but for those interested, we are advised that the "OSK'er" Issue #5 includes a review of the MM-1 machine.

ANSWERS. This edition contains answers, thanks to Bob Devries, to questions raised by some members and we hope that others may also benefit.

MSDOS to OS9 File transfer. We have received a request for an OS9 utility to read and write MS-DOS files. The archive "pcdos.ar" from the P.D. library contains a programme that will do this. It will only work on a 40 track drive (an 80 track drive switchable to 48 T.P.I. is O.K.) and needs SDISK3 or a patched CC3Disk driver. This utility does not handle sub-directories on MS-DOS disks and requires 360k formatted disks. Of course it is only useful on ASCII files (DOS-TEXT).

TSEDIT - TSWORD. These Level 1 utilities will not run "as is" under OS9 Level 2. There is a patch for TSEdit in the P.D. "tsedit.ar". This "Ipatch" file will patch TSEdit so that it will run in a window. Ipatch will also be required. We do not know of such a patch file for TSWORD. Perhaps somebody could help here.

Somebody wants a FORTRAN compiler for the CoCo3. I guess there must be a good reason for this request but we do not know of such a programme. Can anybody help?

NEWSLETTER CONTRIBUTIONS. It is pleasing to note that a number of members have indicated that they CAN contribute articles or questions for inclusion in our newsletter. We look forward to receiving anything in this line.

Cheers, Gordon.

Basic09 Tutorial
by Bob Devries

I decided that I would digress from my tirade about the CGFX2 library calls to explain the answer to a problem that was posed by one of the Brisbane members of the Usergroup. I can see he has a red face. He asked me if I knew of a programme to read text file originating from MS-Dos WordStar or Word-Perfect -and the like-word processors. These word processors set the eighth bit in some characters in the file. I said to him, 'why don't you write one in Basic09?'. He gave me a queer look.

The problem is really not very difficult. What is needed to change the character from high-bit-set to normal? Here's a bit representation of a character 'B' ... 01000010. If the high bit (bit 7, starting at 0) then the bit pattern is like this... 11000010. So the hexadecimal goes from \$42 to \$C2, and the decimal from 66 to 194.

To get rid of that extra bit, we use the Basic09 LAND operator. Land does a Logical AND operation between the two numbers passed in its parameter variables. The first parameter is the character you want to alter, and the second is the mask to be ANDed with. First, let's look at how binary and works. When two binary bits are ANDed together, the result only if BOTH the bits are set to 1. So in our example numbers here's what happens:

```
$C2    11000010
$7F    01111111
-----
```

result: 01000010 which is 42, the value we really want.

Notice how we made sure the bit we want to discard is left as a 0 in the mask, hence the \$7F.

OK, so now the easiest way to filter the

files required is to read and write standard in and standard out, and use input and output redirection. In Basic09, that is easy. Here is the code:

```
PROCEDURE filterhibit
0000   DIM a:STRING[1]
000C   DIM b:BYTE
0013   WHILE NOT(EOF(#0)) DO
001D     GET #0,a
0026     b=ASC(a)
002F     b=LAND(b,$7F)
003B     a=CHR$(b)
0044     PUT #1,a
004D   ENDWHILE
0051   END
```

See, only ten lines of code! First, dimension an input/output variable, 'a' as a single character string, and a byte variable 'b'. Remember, we only need 8 bits, so a BYTE value is required. We then use a while/endwhile loop to read path 0 (standard in) until an EOF (End Of File) is detected. Read the character into 'a', convert to ascii value in 'b', Logical AND 'b' with \$7F, and change it back to a character value with CHR\$. Lastly, PUT it out on path 1 (standard out), and let OS9 worry about where it goes. When EOF is detected, we quit. That's all there is to it.

So you can see that Basic09 is really easy to use for even small tasks like this. The command line for using this programme is this:

filterhibit <msdos.txt >os9.txt

Note that you can only use it once it has been PACKed by Basic09, and stored in your CMDS directory. Well, thats all for now.. keep trying those little programmes in Basic09.

Regards, Bob Devries

oooooooooooo0000000000oooooooooooo

CoCo-Link

CoCo-Link is an excellent magazine to help you with the RSDOS side of the Colour Computer. It is a bi-monthly magazine published by Mr. Robbie Dalzell. Send your subscriptions to:

CoCo-Link
31 Nedlands Crescent
Pt. Noarlunga Sth.
South Australia
Phone: (08) 3861647

The OS9 File System
by Bob Devries

How does OS9 find the files on its disk devices? How does it know what the file's length is? How does it find what free space there is on a disk? How does it know which filenames are files, and which are directory names?

I suppose we have all asked these questions at one time or another. I believe that there are very few people who actually know how their operating system stores files, and how it retrieves them. Well, I'm here to tell you about the way OS9 and OS9/68000 does it.

First of all, you should all know that OS9 can handle any size disk drive up to a maximum of 65535 bytes in its sector allocation table. There are eight bits in a byte, so eight clusters per byte of allocation table are allowed. A cluster is one or more sectors grouped together and taken as one unit. Colour Computer OS9 usually uses one sector clusters, so a disk is limited to 65535 times 8 sectors. With the format of the Colour Computer OS9 disk being eighteen sectors per track, that would make 29126 tracks, and on double-sided disks 14563 tracks. At the moment, I don't know of any disk drives of that number of tracks. Even hard disk drives, which usually have 32 or 46 sectors per track, that would make 16383 tracks or 11397 tracks. My 20 Meg hard drive has 615 tracks, and four sides (two disks) and 32 sectors per track which makes 78720 sectors

OK, so how does OS9 know what type of disk you are using? The answer is: by looking in the device descriptor in memory, which has the data in it (among other things) to tell it how many tracks, sides, and sectors per track your drives have. That way it is easy to change hardware, such as from 40 track to 80 track drives. Just plug them in, change the device descriptor, and go! OS9, is however, even smarter than that. It will allow you to insert a single sided 35 (or even less) track disk in an 80 track drive, and it will adjust itself accordingly.

So where does it get the information? From the first sector of the disk on side 0. OS9 calls that sector LSN0 (for Logical Sector Number 0). The first three bytes of that sector give the number of sectors on the disk, and at byte 16, 17 and 18 are the disk format (density, number of sides) and number of sectors per

track. So OS9 looks at that information and works out if it can read it with the current device descriptor. If the format is less than or equal to that descriptor, it reads it, otherwise, (if you put an 80 track disk in a drive with only 40 tracks) it returns an error 'Wrong Type'.

So once OS9 has that information, it reads another three bytes of information from LSN0 (bytes 8,9 and 10) which tell it the sector number of the ROOT DIRECTORY. Actually, those bytes point to the file descriptor for that directory, not the directory itself. When it has that information, it reads from that sector the sectors used by the root directory. In OS9, each directory may be any length, and grows dynamically as needed (including the root directory, unlike some other systems including MS-DOS). So the file descriptor has in it the pointers to the sectors where the directory information resides on the disk.

So now we come to the actual directory entries themselves. Each directory in OS9 consists of a 29 character filename (last character has 128 added to its ASCII value) and a 3 byte pointer to the sector which contains the file descriptor for that entry. The first entry is a pointer to its own file descriptor, and the second entry is a pointer to the parent directory. In the case of the root directory, these point to the same sector, since the root directory does not have a parent directory. After these two entries come the filename entries or other directory names. The directory entry does not show if the name is a file or a directory name. This information is in the file descriptor, the first byte. If bit seven of that byte is set (to one), that entry is a directory, else it is a file. Directories are treated just like files in every other way, however, and may be read to see what is in them, provided they are opened with the correct permissions (we'll look at that later).

Deleted files have the first byte of their filename set to zero, and have the sector list and length in the file descriptor set to zero. The directory's file descriptor is updated every time it is lengthened by adding a new file to it, but it is not shortened. OS9 itself (the RBF or Random Block File manager) takes care of looking for unused directory entries to fill.

When OS9 has found the filename you wanted it to open, it looks first at the three byte sector number which points to the file descriptor, then in the file descriptor there are up to 48 segment pointers which allow for file fragmentation. The 48 segment pointers are made up of a three byte pointer to the first sector which contains the data of the file, and then a two byte value which is the number of sectors contained in that segment. Let's look at an example:

```
0440 4F5339426F6FF4F4 OS9Boott
0448 0000000000000000 .....
0450 0000000000000000 .....
0458 000000000000000B .....
```

This is a part of my boot disk. Notice the second 't' in OS9Boott. It is just a character left over from a previous name in that directory entry, which has been deleted, and subsequently overwritten. Of course 'OS9Boott' is only seven characters long, and most of the directory entry is filled with zeroes. The last three bytes, the sector pointer to the file descriptor of the file OS9Boott, make up the value \$00000B which is sector eleven. Let's look at the file descriptor:

```
0B00 0300005B08011518 ...[...
0B08 010000787A5B0801 ...xz[...
0B10 00000C0079000000 ...y...
0B18 0000000000000000 .....
0B20 0000000000000000 .....
0B28 0000000000000000 .....
0B30 0000000000000000 .....
0B38 0000000000000000 .....
0B40 0000000000000000 .....
0B48 0000000000000000 .....
0B50 0000000000000000 .....
0B58 0000000000000000 .....
0B60 0000000000000000 .....
0B68 0000000000000000 .....
0B70 0000000000000000 .....
0B78 0000000000000000 .....
0B80 0000000000000000 .....
0B88 0000000000000000 .....
0B90 0000000000000000 .....
0B98 0000000000000000 .....
0BA0 0000000000000000 .....
0BA8 0000000000000000 .....
0BB0 0000000000000000 .....
0BB8 0000000000000000 .....
0BC0 0000000000000000 .....
0BC8 0000000000000000 .....
0BD0 0000000000000000 .....
0BD8 0000000000000000 .....
```

```
0BE0 0000000000000000 .....
0BE8 0000000000000000 .....
0BF0 0000000000000000 .....
0BF8 0000000000000000 .....
```

The first byte is the file's permission bits. In this case, only owner read and owner write are set. The next two bytes are the owner's user ID (usually 0). The next five bytes are the file's last modification date, one byte each for year, month, day, hour and minute. The next byte is the link count (I must look that one up elsewhere... it is not explained in the manual). It seems to be always one. The next four bytes are the file size, in bytes. Next comes the creation date in year, month and day format, one byte each.

The next 240 bytes are made up of a maximum of 48 five byte segments which each contain a three byte pointer to a block of sectors of the file, and a two byte value for the number of sectors in that block. If a file gets fragmented badly enough, you may get error 217 - Segment list full. This means that all 48 of the block pointers have been used. The best way to repair this error, is to copy the file to another disk. This will allocate the file differently, and will remove the problem. Personally I have never struck this problem.

So, to access the file whose file descriptor is displayed above, OS9 reads the length from bytes nine and ten, which gives a value of \$787A or 30842 and then gets the file position pointer from bytes 16, 17 and 18, which gives \$00000C which is sector 12. The next two bytes are the number of sectors (actually clusters) which contain the file, which gives \$0079, 121 decimal. Because the next three bytes are zeroes, OS9 knows that no more clusters are being used. The three byte pointer is converted to a four byte (long) value and is passed to the system call which reads the sector data. The number of bytes to be read from the last sector is worked out by multiplying the two byte segment value (121) by 256 (the number of bytes in a sector) and subtracting the file size (30842). This leaves 134 bytes to be read from the last sector. And voila! the End Of File is reached.

Well, I hope you all made something from that. I know I learnt something from looking it all up. Until next time...

Regards, Bob Devries.

AUSTRALIAN OS9 NEWSLETTER

Utime.h header file for kreider library

It has been brought to our attention that there is a problem in the Public Domain library archive file 'header.ar' on disk 4. The archive contains several header (.h) files which are needed for the C library from Carl Kreider. The header file in question, the last one in the archive, is utime.h. It appears that the last few lines are missing. Here is the complete file in its correct form:

```
/*
**    Utime.h
*/

struct tm {
    int    tm_sec;        /* seconds (0 - 59)    */
    int    tm_min;        /* minutes (0 - 59)    */
    int    tm_hour;       /* hours (0 - 23)      */
    int    tm_mday;       /* day of month (1 - 31) */
    int    tm_mon;        /* month of year (0 - 11) */
    int    tm_year;       /* year (year - 1900)  */
    int    tm_wday;       /* day of week (Sunday = 0) */
    int    tm_yday;       /* day of year (0 - 365) */
    int    tm_isdst;      /* NOT USED            */
};

long      time();        /* Same as UNIX */
struct tm *localtime(); /* Same as UNIX */
char      *asctime();    /* Same as UNIX */
char      *ctime();      /* Same as UNIX */

long      o2utime();     /* Convert OS9 style buf to UNIX long */
/* Void */ u2otime();    /* Convert 'tm' to OS9 char *buf */
```

Rest assured that we will be changing the library file on the PD disk to remove any further cause for concern. This problem caused some consternation with local group members who tried to compile a C programme which called the function o2utime() which, because it was not declared, returned an integer value instead of a long as it should. Hope this clears up some problems.

oooooooooooo0000000000oooooooooooo

68xxx Machines A new magazine for Motorola processors

At the beginning of this year, a new magazine was started in USA by Mr Jim DeStafeno, called 68xxx Machines. It is a magazine devoted only to Motorola processors, and in particular to the 68000, 68010...020 etc. Thus it covers OS9/68K, and various operating systems including SK*DOS. Articles presented cover Basic09, C, assembler, as well as general interest subjects.

Subscription rates to Australia are:

1 Year (surface mail)	US\$17.25
2 Years " "	US\$33.50
1 Year (airmail)	US\$22.50

All major credit cards are accepted, but attract a 10% surcharge. The address is:

The 68xxx Machines
RD 1, Box 375
Wyoming DE 19934
USA

oooooooooooo0000000000oooooooooooo

SMILE! YOU'RE ON CANDID CAMERA!

Date: 12-24-88 09:41
From: Werner Cappel
To: Ron Dwight
Subj: Re: Jokes, Smiles and Clarity

Here's the file :

:~) Joking or sarcastic face
:-} Fiendish grin
;-) A wink
:-(Sad or angry face
!-(Late night
~/~ Stirring up trouble

:~) the normal smiling face, appended to a sentence or an article means 'this is a joke' or 'this is supposed to make you laugh'

:~> normal smiling face with deformed lips, same as above except person who submitted it has problems with their lips

:~{} normal smiling face with a mustache, same as #1 except submitter has mustache

:~} normal smiling face with pretty lips, same as #1 except person that submitted it is wearing lipstick or some other lip appearance improving device

:~! no expression face, 'that comment doesn't phase me'

:~(sad face, 'that comment makes me sad [mad]'

;~) smiling face gets his lights punched out (could be pirate smiling face??), submitter is a practical joker who played one too many and got beat up

;~! no expression face gets his lights punched out, says nothing but still gets beat up

;~(sad face gets his lights punched out, sad or mad and got beat up, or 'that makes me so mad that if I ever see you I'll punch your lights out'

:~\ popeye smiling face, for people who look like popeye

;~\ popeye gets his lights punched out

:~} biting sarcasm smiling face, used when sarcasm is intended, since we cannot inflect our voice over the net

:~{ biting criticism smiling face, ditto for criticism

:*) drunk smiling face, for those of us who like get intoxicated before or while reading netnews

:) smiling face needs a nose job, no explanation necessary

: submitter is a robot (or other appropriate AI project)

:>) submitter has a big nose
:<| submitter attends an Ivy League school
:%}% submitter has acne
=-:-) submitter is a hosehead
:-(* submitter is getting sick of most recent netnews articles and is about to vomit
:-)8 submitter is well dressed
8:-) submitter is a little girl
:-)-{8 submitter is a big girl
%-) submitter is cross-eyed
#-) submitter partyed all night
:-* submitter just ate a sour pickle
:-:-) submitter sports a mohawk and admires Mr. T
(:)-) submitter likes to scuba dive
:-'| submitter has a cold
:-)' submitter tends to drool
':-) submitter accidentally shaved off one of his eyebrows this morning
(-: submitter is Don Ellis from Tektronix
8:] normal smiling face except that submitter is a gorilla
%- submitter is Picasso
(-: submitter is left-handed
0-) submitter is an arc-welder
*:o) submitter is a Bozo
8-) submitter wears glasses
*(<:-) submitter is Santa Claus (Ho Ho Ho)
o-) submitter is a cyclops
%-) submitter with long bangs
(-) submitter needing a haircut
0-) submitter wearing scuba mask
*-(Cyclops got poked in the eye
%-) person submitting has been staring at the terminal for 36 hours
B-! person submitting is wearing cheap sunglasses
P-) person submitting is getting fresh
:'(submitter has had his nose put out of joint. Useful for replying to flames.
!-) Submitter is asleep (boredom)
.-) Submitter has one eye
'-) Submitter only has a left eye, which is closed
:-) Submitter has two noses
:-D Submitter talks too much
:-o Submitter is shocked
:-{ submitter has read too many of the toilet paper articles previous to lunch
:-% Respondent has beard.
:-@ Respondent's beard has permanent wave
or was drawn by Picasso.
:-# Respondent's lips are sealed.
-(:-) Submitter is the pope
:-) respondent wears glasses

```

:-} respondent wears lipstick
{(:-) Submitter is wearing toupee.
}(:-) Submitter, wearing toupee in wind.
<(:-)= Submitter is Chinese. (sorry bout that)
<(:-)= Submitter is Chinese and doesn't like
this article.
:-( Submitter has read too many 'smiley'
articles.
:-) ha ha
:-( net.flame
(:-) hee hee
0!-) net.religion
!-D ho ho
:-> hey hey
8:-I net.unix-wizards
:-( boo hoo
X-( net.suicide
:-I hrm
E:-I net.ham-radio
:-O uh oh
>:-I net.startrek
:-P nyah nyah
3:o[ net.pets
!-P yuk
:-} beard
:-{ mustache
:-# braces
:-X bow tie
:-Q smoker
<:I dunce
(:I egghead
8-) glasses
B-) horn-rims
8:-) glasses on forehead
:-8( condescending stare
;-) wink
:-< mad
:-` smiley spitting out its chewing tobacco
:-l smiley bland face
:-! "
:-@ smiley face screaming
:-# smiley face with bushy mustache
:-$ smiley face with it's mouth wired shut
:-& smiley banker
:-6 smiley after eating something sour
:~) smiley with pointy nose (righty)
:-7 smiley after a wry statement
8-) smiley swimmer
:-* smiley after eating something bitter
:-& smiley which is tongue-tied
:-9 smiley licking it's lips
:-0 smiley orator
:-( un-smiley
:-) smiley standard

smiley invisible man
(:-( unsmiley frowning
(:-) smiley big-face
):-) "
):-( unsmiley big-face
)8-) scuba smiley big-face
=-(-) smiley punk-rocker
=-(-) (real punk rockers don't smile)
=-#) smiley punk with a mustache...
:-) smiley priest
:-q smiley trying to touch its tongue to its
nose
:-e disappointed smiley
:-t cross smiley
:-i semi-smiley
:-o smiley singing national anthem
:-p smiley sticking its tongue out (at you!)
:-{ un-smiley blockhead
:-} smiley blockhead
:-{ smiley variation on a theme
:-} ditto
{(:-) smiley with its hair parted in the middle
}:-) above in an updraft
:-a lefty smiley touching tongue to nose
:-s smiley after a BIZARRE comment
:-d lefty smiley razzing you
g-) smiley with pince-nez glasses
:-j left smiling smiley
:-k beats me, looks like something, tho.
:-l y. a. s.
:-: mutant smiley
:-\ undecided smiley
:-{ "have an ordinary day" smiley
;-) winking smiley
:-< real sad smiley
:-> y.a.s.
:-z y.a.c.s.
:-x "my lips are sealed" smiley
:-c bummed out smiley
:-v talking head smiley
:v) left-pointing nose smiley
:-b left-pointing tongue smiley
:-/ lefty undecided smiley
:-? smiley smoking a pipe
.-] one-eyed smiley
,-{ wry and winking
0-) smiley cyclops (scuba diver?)
:-=) older smiley with mustache
:u) smiley with funny-looking left nose
:n) smiley with funny-looking right nose
:< midget unsmiley
:> midget smiley
:-)/2 remark was only half in jest "We laugh
because it hurts so much"

```

oooooooooooooooooooooooooooooooo

Some Notes from the InterNet OS9 area

C Compilers, and 68K machines
- James Jones

Recompiling C code from CoCo 3 source to machines such as the MM/1 will require some alterations. One case in point is what to use in place of the os9.h file, for use on the 68K platforms. You may recall, that os9.h contains a declaration of a structure that corresponds to the registers of the 6809.

Such a structure wouldn't do much good on a 680xx! Even if one wrote something that corresponded to the 680xx's eight data and eight address registers, the code containing the #include would certainly have to change.

Although there is a syscall for Microware BASIC (the 680xx equivalent of BASIC09), it, too, has to have a different TYPE statement, and references to that have to change correspondingly. (One of the few sources of incompatibility between BASIC09 and Microware BASIC.)

The C library for OS-9/68000, on the other hand, has gone the route of having individual functions corresponding to system calls instead of the catcall os9(). People who have been using the Carl Kreider library under OS-9/6809 win big here, because his library takes the approach used in OS-9/68000, and changes to code using his library should be minimal for a port to OS-9/68000.

More BLOB's
- Rick Adams

My explorations into the BLOB problem came about when I noticed a BLOB of my own in a piece of RSDOS code. Normally, this is very rare, but I found that modification of DELPHIterm's code would sometimes cause its disk drivers to become flaky, dropping a byte on a sector read or write about one time out of every 20 reads.

Experimentation showed me that I could fix the problem, if it occurred, by inserting or deleting two NOPs just ahead of the disk driver code. It wasn't a subtle timing change due to the time it took the NOPs to be executed, because they weren't executed -- they were just "spacers" used to move the disk driver code down two bytes. This ALWAYS fixed the problem.

Further experimentation showed that the specific part of the code that was position-sensitive was the read and write sector loops. These loops are very tight, tricky code in which bytes are transferred to and from the disk as fast as the CPU will allow, with no time to even do a "end of loop" test to get out of the loop -- that is handled by an NMI from the disk controller that kicks the CPU out of the loop and into a "end of sector" routine.

Successive tests showed that the loop would be flaky on one out of every four addresses... if it was bad, I could fix it by moving it down one, two, or three bytes. But if I moved it down or up by four, the problem would remain.

I speculated then that the OS9 BLOB could be reliably conquered via the use of a small "spacer" module that did nothing but move the OS9 disk driver module down just a smidgeon... as long as the length of the module was NOT divisible by four. :-)

For some reason, I've never had a BLOB in my boot (tho I *do* use OS9 extensively, and fiddle with my boot now and then), so I've never been really motivated to try this.

More News on CoCo BLOB's
- Kevin Darling

I suppose it's okay to mention something now. Years ago, Tandy knew about the BLOB, at least as related to bad disk reads under RSDOS. That was something I'd never heard of, but apparently they had. So they had Western Digital spend weeks trying to track down the problem, but it was never found.

I think (not sure) that WD subcontracted the microcode, and that didn't help either. Western Digital didn't have the source or something. In any case, people have been chasing this sucker for quite some time now, if that helps any <grin>. It may be that many computers had similar problems, but we were just more likely to notice them (since we expect our machines to be perfect :-).

It has also been mentioned that "E-clock gating" was a possible fix, which was recommended earlier by C. Bundy on the CoCo Internet List. However, on systems using a Disto SC-II, this fix is included in the disk controller board.

Perhaps other systems could try it.

This can be done by jumpering the expansion bus pin 6 to pin 40 inside the CoCo. That ties the E clock to the SLENB line on the 138 chip. Do not try this fix unless you understand exactly what you are doing.

NEC 3D Monitor (-) CoCo 3
- Tim Kientzle

Perhaps it was a silly thing, but somewhere in the process of thinking about buying a new computer, I ended up with a monitor before I got the computer. Thinking that a "good" Multisync would be generally useful for any computer I would be likely to get, I jumped at a good price on the NEC 3D.

Now, I'm trying to connect this beast to my CoCo3, and the lack of success is making me doubt my decision. Does anyone know how to do this?

P.S. Predictably, NEC customer support has been less than helpful so far. But, it is a free call, and I'm perfectly willing to keep bothering them until I get a useful answer. <grin> (Something other than "Is that CGA, EGA, or VGA??" <sigh>)

Answer 1
- Charles C. Bundy

Grab a female 9 pin sub-D connector. Make a one-to-one connection from the CoCo's RGB connector to the 9 pin. (Note pin 10 from the CoCo isn't used.) You can find the pin numbers marked on the front/back of most 9 pin sub-D's. I'm not sure of the CoCo orientation, but the numbers are marked on the motherboard. Make sure 1 goes to 1, 2 goes to 2, etc. Don't worry about the sound pin (7, I think) it goes to an equivalent NC on the 9 pin side (CGA/EGA pinout standard 7NC). Once you have your 9 pin connector you'll need to plug the HD 15 pin adapter in to the NEC 3D side. (I hope you got one of these with the monitor! Plug that into your 9 pin and you're set. I've used a Relysis and NEC 3D MultiSync monitor on my tower case CoCo - works beautifully. (The NEC 3D won't need any switch settings but the Relysis required the auto/manual switch be set to manual and the TTL/analog switch set to TTL). Yes, TTL! All 16 colors showed up while playing Sinistarr, so I'm assuming it was a frequency dependent setting,

not a signal dependency. When set to analog I got little blue dots, but no picture...

The Nec 3D unlike the less expensive Relysis will do configuration with no switch settings. The mode/color switches are used to fine tune (or override) what the Nec 3D has determined about your video display. In the case of the CoCo III try Mode switch OFF, Color Switch AUTO. Regardless, you should get more than a blank display no matter what those switches are set to.

Make sure all ground pins are tied together on the 15 pin HD connector. These pins should go to pins 1 and 2 on the CoCo connector. The pins are 4, 5, 6, 7, 8, 10, 11 (6=R Gnd; 7=G Gnd; 8=B Gnd; 4, 5, 10, 11 Gnd).

I would still recommend going with a CGA connector on the CoCo and using the 15 to 9 pin adapter. This has worked for me.

Answer 2
- Ken Scales

The limited info I have for the NEC 3D gives the following pinouts for Analog connections with separate Hsync and Vsync:

Monitor Pin Number	Signal	Coco Pin Number
1	Red	3
2	Green	4
3	Blue	5
4	H Sync	8
5	V Sync	9
6	Red Ground	(1)
7	Green Ground	(1)
8	Blue Ground	(1)
9	Ground	1

(Of course, the third column is my own addition -- NEC left this info out of their manuals <grin>).

Note these pinouts are COMPLETELY different from those for a CGA-type of connection, which would have the monitor pins connected to the same-numbered pins on the Coco -- that arrangement will give you only 8 colors.

Looks like the NEC uses similar methods to the PanaSync for switching between the different modes: checking for Sync signals on certain pins which would indicate whether it should operate

in TTL/CGA/EGA (sync on pins 8 and 9), PGC Analog (composite sync on pin 4), or VGA/MCGA/MACII Analog (H sync on 4, V sync on 5). (Actually that last part was a bit of a simplification, but covers the highlights.)

Delays Everywhere!
- Kevin Darling

[Thought y'all would find this amusing, since it reminds me of some Internet postings I've read lately about the new OSK machines...]

"This is the way that it is SUPPOSED to work, but unfortunately it isn't. As somebody said at a user group meeting Saturday 'The only thing [Company X] has shipped lately is excuses'.

There have been numerous complaints here from individuals who have been waiting 4-6 weeks for machines and many of those that have got their [machines] can't get a newer [version of windows]. Other users mentioned at our meeting that [Company X's] 1-800 number is not being answered either these days.

Sorry to vent so much steam. Even though I have my [machine] and can live with [current software] for the time being, I feel real sorry for friends that have heard excuse after excuse this summer.

First in late May it was that there is a backlog and it will take 2 weeks to get a machine in. Early June it was "[Company X] shut down for a week around Memorial day. Shipping will start Monday". About 2 weeks ago it was "The machines will be shipped on Friday, and will take about 3 or 4 days to get here."

Last week it was 4-8 weeks, at the earliest.

What more can I say!"

So who is the "Company X" that's being ripped here? In this case, Commodore. And the machine is the Amiga 3000. But you can find the same thing in Atari, NeXT, and even Apple forums.

That's just the way it sometimes goes.

MM/1 News
- Mark Griffith

From a recent IMS announcement, it looks like the MM/1 will be able to send and receive fax images!

That fax deal is just a sendfax modem and some software to send text files to the modem to send out as a fax. It can't receive fax images. I'm sure someone will do that in the near future.

What are people are contemplating for a backup medium for their MM/1's. A 105Mbyte Hard Disk is going to take a lot of floppies, even 1.44 Mbyte ones.

Carl Kreider has been working on some tape drivers for a couple different tape units. As soon as he is finished with his current projects (the MM/1 SCSI driver and new floppy driver), he will probably get back to the tape units. He does have the drivers working on Atari ST machines, so it should be pretty easy to port to the MM/1. The only problem is going to be finding a fairly cheap SCSI tape drive that is also reliable. The TEAC units are darn good, use those audio cassette looking tapes, store up to around 60 or 70 meg, but aren't cheap.

Some good news is future MM/1 designs will support the newer 4 Meg floppy disks. Only a couple minor changes need to be made in the PCB to handle it.

What is in the mysterious Level 2 upgrade?
- Mark Griffith

A new ACIA driver. Very Similar to Bruce Isted's SACIA driver. (Which is available from him directly along with his Eliminator board, and has the same features as the upgraded Level II version). I'm making an educated guess that the two drivers are one-in-the-same, or at least functionally the same.

This driver required a new clock module to work at its best, which was included in the Level II upgrade.driver.

As well as a better serial driver, it included new kernel modules that changed the way OS-9 tasks switches making it all much smoother and causing most processes to run slightly faster...or at least appear to do so. There were also small changes to fix known bugs that had been there for years.

BBF was changed to allow undeleting a file or a whole directory. There is even an UNDELETE utility included.

The GFX2 file most BASIC09 programmers are familiar with was a result of the upgrade. At one time, vector fonts were supported and I believe Hershey Fonts were being worked on at one point. A neat little demo was made that put up on the screen (real fast I might add) fancy cursive characters that said:

"This is NOT your Father's OS-9"

These fonts were dropped though!

Support is in there for moveable, overlapping windows too. Pretty fast and easy to use. Move the mouse over a window and press the select key (yes, the second button on the mouse was used for window select) and that became the active window. You could even update the window in the background as you used the one in the foreground (not nearly as slow as you might think).

I'm sure there are other things in there I can't remember now. It was certainly worth the effort for the people involved to do it. Too bad Tandy killed it before it could be released.

Backing up hard disks

Peter Edwards asks:

Now that I have my hard disk running on my CoCo, I have to figure out how to back it up.

I have tried BRU, from Delphi, and it worked well, but took about 15 minutes per 360k floppy. Pretty slow, especially if you compare it to the time Fastback+ can dump IBM disks. (IMHO, if you really MUST use a PC clone, you GOTTA have Fastback+.) Wonder if BRU would benefit from a different interleave on its floppies? Hmm...

I also have the Burke & Burke HDB/HDR programs, and the HDKIT suite from the CoCo Internet List, although I haven't tried them yet (have to go buy more floppies first!)

What do people use, and are there any comments on the three mentioned above? Are there any others around?

Elena Marcias Replies:

I use HDB from Burke & Burke for my 30 Meg drive. It takes about 20 720K disks to do a complete backup so I speed things up by turning off the write verify on the floppy drive to which I write. I do this by running:
dmode /f2 vfy=01 (f2 is the drive I backup to).

This way each disk takes only 3 minutes! To be on the safe side, once a year I backup up all my files with AR so I can restore them in case any of the HDB disks are damaged or contain errors.

oooooooooooo0000000000oooooooooooo